



TITLE:

# Learning Equal Matrix Grammars and Multitape Automata with Structural Information

AUTHOR(S):

Takada, Yuji

---

CITATION:

Takada, Yuji. Learning Equal Matrix Grammars and Multitape Automata with Structural Information. 数理解析研究所講究録 1990, 731: 61-72

ISSUE DATE:

1990-10

URL:

<http://hdl.handle.net/2433/101978>

RIGHT:

# Learning Equal Matrix Grammars and Multitape Automata with Structural Information

高田裕志  
(Yuji Takada)

*International Institute for Advanced Study of  
Social Information Science (IIAS-SIS)*  
FUJITSU LIMITED

140, Miyamoto, Numazu, Shizuoka 410-03, Japan  
E-mail : yuji%iias.fujitsu.co.jp@uunet.uu.net

## Abstract

Equal matrix grammars are one of parallel rewriting systems. This type of parallel rewriting systems has been investigated in several areas such as  $L$ -systems and the syntactic pattern recognition. In this paper, we introduce two types of structural information and show that if these are available and there is a method of learning regular sets in polynomial time, then for any positive integer  $k$  the class of equal matrix grammars of order  $k$  is learnable in polynomial time. We also show that for any equal matrix language there exists an equal matrix grammar learnable efficiently only from positive structural examples. These results are applied to the problem of learning multitape automata and the same results are obtained.

## 1 Introduction

Equal matrix grammars introduced by Siromoney [7] are one of parallel rewriting systems. This type of parallel rewriting systems has been investigated in several areas such as  $L$ -systems ( $n$ -parallel right linear grammars [11]) and the syntactic pattern recognition [6]. Also, they are one of general extensions of regular grammars and closely related to multitape automata, which are one of generalizations of finite automata. These observations lead us to the study of learning method for equal matrix grammars and languages.

At first, we consider a relation between the learning problem for equal matrix languages and the problem for regular sets. Then, we introduce two types of structural information on equal matrix grammars and define two subclasses of the grammars for each order  $k$  ( $k$  is a positive integer). It is shown that if there is a method of learning regular sets in polynomial time, then for each order  $k$  these subclasses are learnable in polynomial time with the method. This implies that if the structural information on grammars is available then the class of equal matrix grammars of order  $k$  is learnable efficiently. We also show that for any equal matrix language there exists an equal matrix grammar learnable efficiently

only from positive structural examples. These results are applied to the problem of learning multitape automata and the same results are obtained.

## 2 Preliminaries

Let  $\Sigma$  denote an alphabet and  $\Sigma^*$  denote the set of all strings over  $\Sigma$  including the null string  $\lambda$ .  $lg(w)$  denotes the length of a string  $w$ . A *language* over  $\Sigma$  is a subset of  $\Sigma^*$ .

A *finite automaton* over  $\Sigma$  is defined as usual and denoted by a 5-tuple  $M = (K, \Sigma, \delta, q_0, F)$ , where  $K$  is a set of states,  $\delta$  is a transition function,  $q_0$  is the initial state, and  $F$  is the set of final states. We abbreviate a deterministic and nondeterministic finite automaton as *DFA* and *NFA* respectively. A *regular set*  $R$  is a subset of  $\Sigma^*$  accepted by an *NFA*.

Let  $k$  be a positive integer. An *equal matrix grammar of order  $k$* , abbreviated *k-EMG*, is a  $(k+3)$ -tuple  $G = (N_1, \dots, N_k, \Sigma, \Pi, S)$ .  $N_1, \dots, N_k$  are finite nonempty pairwise disjoint sets of *nonterminals*.  $S$  is not in  $N_1 \cup \dots \cup N_k \cup \Sigma$  and is called the *start symbol*.  $\Pi$  is a finite nonempty set of the following three types of *matrix rules*:  $[S \rightarrow u_1 A_1 \dots u_k A_k]$ ,  $[A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k]$ , and  $[A_1 \rightarrow u_1, \dots, A_k \rightarrow u_k]$ , where for each  $i$  ( $1 \leq i \leq k$ ),  $A_i, B_i$  are in  $N_i$  and  $u_i \in \Sigma^*$ . Especially, a *k-EMG*  $G^0$  is said to be *minimal* if and only if for any  $i$ ,  $N_i$  is a singleton, i.e.,  $N_i = \{S_i\}$ .

In what follows, we denote any matrix rule with its unique label  $\pi$ . The *size* of a matrix rule  $\pi$ , denoted  $size(\pi)$ , is defined as follows: If  $\pi$  is of the form  $[S \rightarrow x]$  then  $size(\pi) = lg(x)$ . If  $\pi$  is of the form  $[A_1 \rightarrow x_1, \dots, A_k \rightarrow x_k]$  then  $size(\pi) = \max(lg(x_1), \dots, lg(x_k))$ .

Let  $G = (N_1, \dots, N_k, \Sigma, \Pi, S)$  be a *k-EMG*. We denote  $N_1 \cup \dots \cup N_k \cup \Sigma \cup \{S\}$  by  $V$ . For any  $x, y \in V^*$ ,  $x \xrightarrow[\pi]{\pi} y$  if and only if either  $x = S$  and  $\pi : [S \rightarrow y]$  is in  $\Pi$  or there exist  $u_1, \dots, u_k, v_1, \dots, v_k$  in  $\Sigma^*$ ,  $z_1, \dots, z_k$  each  $z_i$  in  $(N_i \cup \Sigma)^*$ , and  $A_1, \dots, A_k$  each  $A_i$  in  $N_i$  such that  $x = u_1 A_1 v_1 \dots u_k A_k v_k$ ,  $y = u_1 z_1 v_1 \dots u_k z_k v_k$ , and  $\pi : [A_1 \rightarrow z_1, \dots, A_k \rightarrow z_k]$  is in  $\Pi$ . We write  $x \xrightarrow[\alpha]{\alpha} y$  if and only if either  $x = y$  and  $\alpha = \lambda$ , or there exist  $x_0, \dots, x_n \in V^*$  such that  $x = x_0$ ,  $y = x_n$ , and  $x_i \xrightarrow[\pi_i]{\pi_i} x_{i+1}$  for each  $i$  and  $\alpha = \pi_1 \dots \pi_n$ .  $x \xrightarrow[\alpha]{\alpha} y$  is called a *derivation from  $x$  to  $y$  with an associate word  $\alpha$  in  $G$* . A *k-EMG*  $G$  is said to be *unambiguous* if and only if for any string  $w \in \Sigma^*$ ,  $S \xrightarrow[\alpha]{\alpha} w$  and  $S \xrightarrow[\alpha']{\alpha'} w$  imply  $\alpha = \alpha'$ .

The *language generated by  $G$* , denoted  $L(G)$ , is the set  $L(G) = \{w \in \Sigma^* \mid S \xrightarrow[\alpha]{\alpha} w\}$  and is called the *equal matrix language of degree  $k$*  (abbreviated *k-EML*) generated by  $G$ .

Clearly, any 1-EML is a regular set and it is easy to show that any 2-EML is a linear language. For any positive integer  $k \geq 3$ , the class of *k-EMLs* contains some context-sensitive languages. For example, the context-sensitive language  $\{a^n b^n c^n \mid n \geq 1\}$  is a 3-EML. Also, there exists a context-free language which is not a *k-EML* for any positive integer  $k$  [4]. For example, the context-free language  $\{a^n b^n \mid n \geq 1\}^*$  is not a *k-EML* for any  $k$ . The class of *k-EMLs* forms a hierarchy of languages in the family of equal matrix languages [4].

## 3 Representation Theorem

In this section, we show a representation theorem for *k-EMLs*. The theorem claims that any *k-EML* is generated by a minimal *k-EMG* with a *regular* control set. This suggests a relation between the problem of learning *k-EMLs* and the problem of learning regular sets.

**Definition** Let  $G = (N_1, \dots, N_k, \Sigma, \Pi, S)$  be a  $k$ -EMG and  $A(G) = \{\alpha \mid S \xrightarrow[G]{\alpha} w, w \in L(G)\}$ . Then, a subset  $C$  of  $A(G)$  is called a *control set* for  $G$  and  $L_C(G) = \{w \in \Sigma^* \mid S \xrightarrow[G]{\alpha} w, \alpha \in C\}$  is called the *language generated by  $G$  with the control set  $C$* .

We denote by  $\pi^0 : R(\pi, S_1, \dots, S_k)$  the matrix rule such that all occurrences of nonterminals of  $N_i$  in  $\pi$  are replaced by  $S_i$  for every  $i$  ( $1 \leq i \leq k$ ).

Let  $G = (N_1, \dots, N_k, \Sigma, \Pi, S)$  be a  $k$ -EMG and  $G^0 = (\{S_1\}, \dots, \{S_k\}, \Sigma, \Pi^0, S)$  be a minimal  $k$ -EMG such that  $\Pi^0 \supseteq \{\pi^0 : R(\pi, S_1, \dots, S_k) \mid \pi \in \Pi\}$ . Then we define a homomorphism  $h$  from  $\Pi^*$  to  $\Pi^{0*}$  such that  $h(\pi) = \pi^0$  if and only if  $\pi^0 : R(\pi, S_1, \dots, S_k)$ . Also, we define the NFA  $M = (\{S, q_f\} \cup (N_1 \times \dots \times N_k), \Pi^0, \delta, S, \{q_f\})$  corresponding to  $G$ , where  $q_f \notin N_1 \cup \dots \cup N_k$  and  $\delta$  is defined as follows: (where  $u_i \in \Sigma^*$  for each  $i$ )

$$\begin{aligned} \delta(S, \pi^0) &= \{(A_1, \dots, A_k) \mid \pi \in h^{-1}(\pi^0) \text{ and } \pi : [S \rightarrow u_1 A_1 \dots u_k A_k]\}, \\ \delta((A_1, \dots, A_k), \pi^0) &= \{(B_1, \dots, B_k) \mid \pi \in h^{-1}(\pi^0) \text{ and } \pi : [A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k]\}, \\ \delta((A_1, \dots, A_k), \pi^0) &= \{q_f\} \quad \text{if } \pi \in h^{-1}(\pi^0) \text{ and } \pi : [A_1 \rightarrow u_1, \dots, A_k \rightarrow u_k]. \end{aligned}$$

The following lemma can be proved by an induction on the length of associate words.

**Lemma 3.1** For any  $w \in \Sigma^*$ ,  $(A_1, \dots, A_k) \in N_1 \times \dots \times N_k$ , and  $\alpha \in A(G)$ ,  $A_1 \dots A_k \xrightarrow[G]{\alpha} w$  if and only if  $S_1 \dots S_k \xrightarrow[G^0]{h(\alpha)} w$  and  $q_f \in \delta((A_1, \dots, A_k), h(\alpha))$ .

From definitions,  $S \xrightarrow[G]{\pi} u_1 A_1 \dots u_k A_k$  if and only if  $S \xrightarrow[G^0]{h(\pi)} u_1 S_1 \dots u_k S_k$  and  $(A_1, \dots, A_k) \in \delta(S, h(\pi))$ . Therefore, by Lemma 3.1, for any  $w \in \Sigma^*$ ,  $S \xrightarrow[G]{\alpha} w$  if and only if  $S \xrightarrow[G^0]{h(\alpha)} w$  and  $q_f \in \delta(S, h(\alpha))$ . Hence, we have the following:

**Lemma 3.2** For any  $k$ -EML  $L$ , there exist a minimal  $k$ -EMG  $G^0$  and a regular control set  $C$  for  $G^0$  such that  $L = L_C(G^0)$  holds.

**Lemma 3.3** Let  $G^0$  be a minimal  $k$ -EMG and  $C$  be a regular control set for  $G^0$ . Then  $L = L_C(G^0)$  is a  $k$ -EML.

*Proof.* Let  $G^0 = (\{S_1\}, \dots, \{S_k\}, \Sigma, \Pi^0, S)$  be a minimal  $k$ -EMG and  $M = (N, \Pi^0, \delta, S, F)$  be a DFA which accepts  $C$ . We define a  $k$ -EMG  $G = (\{S_1\}, \dots, \{S_{k-1}\}, N, \Sigma, \Pi, S)$  and a homomorphism  $h$  from  $\Pi^*$  to  $\Pi^{0*}$  as follows: (1) if  $\delta(S, \pi^0) = A$  and  $\pi^0 : [S \rightarrow u_1 S_1 \dots u_k S_k]$ , then  $\pi : [S \rightarrow u_1 S_1 \dots u_k A]$  is in  $\Pi$  and  $h(\pi) = \pi^0$ , (2) if  $\delta(A, \pi^0) = B$  and  $\pi^0 : [S_1 \rightarrow u_1 S_1, \dots, S_k \rightarrow u_k S_k]$ , then  $\pi : [S_1 \rightarrow u_1 S_1, \dots, A \rightarrow u_k B]$  is in  $\Pi$  and  $h(\pi) = \pi^0$ , (3) if  $\delta(A, \pi^0) \in F$  and  $\pi^0 : [S_1 \rightarrow w_1, \dots, S_k \rightarrow w_k]$  where each  $w_i$  is in  $\Sigma^*$ , then  $\pi : [S_1 \rightarrow w_1, \dots, A \rightarrow w_k]$  is in  $\Pi$  and  $h(\pi) = \pi^0$ . By the similar argument in the proof of Lemma 3.2, it is easy to prove that for any  $w \in \Sigma^*$ ,  $S \xrightarrow[G^0]{\alpha^0} w$  and  $\delta(S, \alpha^0) \in F$  if and only if  $S \xrightarrow[G]{\alpha} w$  where  $\alpha \in h^{-1}(\alpha^0)$ . Therefore,  $L = L_C(G^0) = L(G)$  is a  $k$ -EML.  $\square$

We summarize Lemmas 3.2 and 3.3 in the following theorem:

**Theorem 3.4** For any language  $L$ ,  $L$  is a  $k$ -EML if and only if there exist a minimal  $k$ -EMG  $G^0$  and a regular control set  $C$  for  $G^0$  such that  $L = L_C(G^0)$  holds.

From this, we define a  $k$ -EML in terms of a minimal  $k$ -EMG and a regular control set for it.

**Definition** Let  $L$  be a  $k$ -EML. A *primitive  $k$ -EMG* of  $L$  is a minimal  $k$ -EMG  $G^0$  such that there exists a regular control set  $C$  for  $G^0$  with which  $G^0$  generates  $L$ .

## 4 Learning Method Based on Control Sets

We assume that a *learner* is a procedure which (1) gets strings as examples, (2) outputs strings for queries, and (3) outputs grammars as conjectures. Although various learning methods for formal languages have been proposed up to now (see [3], for example), this assumption seems to be general enough to include any known learning protocol. Therefore, without loss of generality, we may assume that a learner for regular sets gets strings, outputs strings, and outputs DFAs, and a learner for  $k$ -EMLs gets strings, outputs strings, and outputs  $k$ -EMGs.

Theorem 3.4 implies that an unknown  $k$ -EML  $L$  can be identified by identifying a primitive  $k$ -EMG  $G^0$  and a regular control set  $C$  for  $G^0$ . From this, we divide tasks of a learner for  $k$ -EMLs into the following two main tasks: (1) identifying  $G^0$  of  $L$  and (2) identifying  $C$  for  $G^0$ . In order to construct a learner for  $k$ -EMLs which carries out the above two main tasks, the following three auxiliary tasks may be needed: (a) converting a string to associate words by parsing in  $G^0$ , (b) converting an associate word to a string by generating in  $G^0$ , and (c) constructing a  $k$ -EMG from a DFA. While for the first auxiliary task a method to convert a string to an associate word and its efficiency seem to depend on primitive  $k$ -EMGs, for the other two auxiliary tasks, there are general methods, which the reader finds in [10]. Their time complexities are bounded by polynomials of  $p$ ,  $q$ , and the length of associate words or the number of states of a DFA, where  $p$  is the cardinality of the set of matrix rules of  $G^0$  and  $q$  is the maximum size of the matrix rules of  $G^0$ .

For any alphabet  $\Sigma$  there exists a minimal  $k$ -EMG  $G^u$  fixed for  $\Sigma$  which is a primitive  $k$ -EMG of any  $k$ -EML over  $\Sigma$  [10]. This implies that a  $k$ -EML can be identified by identifying a regular control set for  $G^u$ . However, for each  $k$ -EML  $L$ , there exist more than one regular control sets for  $G^u$  and it seems to be impossible to fix a regular control set for  $G^u$  effectively. Furthermore, more than non-polynomial number of associate words can be converted from an inout string  $w$  by parsing in  $G^u$ .

Finally, we show a condition on minimal  $k$ -EMGs so that there exists a unique regular control set.

**Lemma 4.1** *Let  $L$  be a  $k$ -EML and  $G^0$  be a primitive  $k$ -EMG of  $L$ . If  $G^0$  is unambiguous, then a control set  $C$  with which  $G^0$  generates  $L$  is regular and unique. Moreover,  $C = \{\alpha \in \Pi^{0*} \mid S \xrightarrow[G^0]{\alpha} w, w \in L\}$ .*

*Proof.* Assume that  $C'$  is another control set such that  $L = L_{C'}(G^0)$ . Since  $G^0$  is unambiguous, for any string  $w \in L$  there exists a unique associate word  $\alpha$  such that  $S \xrightarrow[G^0]{\alpha} w$ . Since  $C$  and  $C'$  are subsets of  $A(G^0)$ ,  $w \in L$  if and only if  $\alpha \in C$  if and only if  $\alpha \in C'$ . Therefore,  $C = C'$ .

Let  $G$  be a  $k$ -EMG such that  $L = L(G)$  and  $\Pi^0 \supseteq \{\pi^0 : R(\pi, S_1, \dots, S_k) \mid \pi \in \Pi\}$  where  $\Pi$  and  $\Pi^0$  are sets of matrix rules of  $G$  and  $G^0$  respectively. By the assumption and Theorem 3.4,

such a  $k$ -EMG exists. Let  $C''$  be a regular control set which the NFA corresponding to  $G$  accepts. Since a control set for  $G^0$  is unique,  $C = C''$ , hence  $C$  is regular.  $\square$

Therefore, for an unknown  $k$ -EML  $L$ , if an unambiguous primitive  $k$ -EMG  $G^0$  of  $L$  is found, the learner has only to identify a unique regular control set  $C$ . In this case, for any string  $w \in \Sigma^*$ , if  $S \xrightarrow[G^0]{\alpha} w$ , then  $w \in L$  exactly means  $\alpha \in C$  and  $w \notin L$  means  $\alpha \notin C$ . If there is no  $\alpha$  such that  $S \xrightarrow[G^0]{\alpha} w$ , then  $w \notin L$ . This implies the following proposition:

**Proposition 4.2** *The problem of identifying a regular control set for an unambiguous minimal  $k$ -EMG is reduced to the problem of identifying a regular set.*

## 5 Learning $t$ -Even Equal Matrix Grammars

In this section, we introduce the structural information called  $t$ -evenness and define a subclass of  $k$ -EMGs, called  $t$ -even  $k$ -EMGs. It is shown that the learning problem for the class of languages generated by  $t$ -even  $k$ -EMGs is reduced to the problem for regular sets in polynomial time.

**Definition** Let  $t$  denote a  $k$ -tuple  $(t_1, \dots, t_k)$  such that each  $t_i$  is a positive integer. A  $t$ -even  $k$ -EMG  $G_t = (N_1, \dots, N_k, \Sigma, \Pi_t, S)$  is a  $k$ -EMG such that each matrix rule in  $\Pi_t$  is of the form (1)  $\pi_S : [S \rightarrow A_1 \cdots A_k]$ , (2)  $\pi_N : [A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k]$ , or (3)  $\pi_T : [A_1 \rightarrow u_1, \dots, A_k \rightarrow u_k]$ , and for each  $\pi \in \Pi_t$  of the form  $\pi_N$  or  $\pi_T$ , there exists a positive integer  $n$  such that  $lg(u_i) = nt_i$  for each  $i$ , where  $A_i, B_i \in N_i$ ,  $u_i \in \Sigma^*$  for each  $i$ .

A  $t$ -even  $k$ -EML is a language generated by a  $t$ -even  $k$ -EMG. We note that from the definition, for any  $t$ -even  $k$ -EML  $L_t$ , for any element  $w$  in  $L_t$ ,  $w$  can be partitioned into  $k$  number of substrings  $w_1, \dots, w_k$  such that there exists a positive integer  $n$  and  $lg(w_i) = nt_i$  for each  $i$ .

Clearly, any  $t$ -even 1-EML is a regular set. The class of  $t$ -even 2-EMLs is included in the class of  $k$ -linear languages described in [1]. For a positive integer  $k > 2$ , the class of  $t$ -even  $k$ -EMLs contains some context-sensitive languages.

At first, we show a normal form of  $t$ -even  $k$ -EMGs without the proof.

**Lemma 5.1** *Let  $t = (t_1, \dots, t_k)$ . For any  $t$ -even  $k$ -EML  $L_t$ , there exists a  $t$ -even  $k$ -EMG  $G_t = (N_1, \dots, N_k, \Sigma, \Pi, S)$  such that  $L_t = L(G_t)$  and each matrix rule is of the form (1)  $\pi_S : [S \rightarrow A_1 \cdots A_k]$ , (2)  $\pi_N : [A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k]$ , or (3)  $\pi_T : [A_1 \rightarrow u_1, \dots, A_k \rightarrow u_k]$ , where for each  $i$ ,  $A_i, B_i \in N_i$ ,  $u_i \in \Sigma^*$ , and  $lg(u_i) = t_i$ .*

Given an alphabet  $\Sigma$  and  $t = (t_1, \dots, t_k)$ , we define the universal  $t$ -even  $k$ -EMG  $G^T = (\{S_1\}, \dots, \{S_k\}, \Sigma, \Pi^T, S)$ , where  $\Pi^T$  consists of the following matrix rules:

$$\begin{aligned} \Pi^T = & \{ \pi_S^T : [S \rightarrow S_1 \cdots S_k] \} \\ & \cup \{ \pi_N^T : [S_1 \rightarrow u_1 S_1, \dots, S_k \rightarrow u_k S_k] \mid u_i \in \Sigma^* \text{ and } lg(u_i) = t_i \} \\ & \cup \{ \pi_T^T : [S_1 \rightarrow u_1, \dots, S_k \rightarrow u_k] \mid u_i \in \Sigma^* \text{ and } lg(u_i) = t_i \}. \end{aligned}$$

It is easy to verify that  $G^T$  is unambiguous and generates any string  $w$  over  $\Sigma$  such that  $w$  can be partitioned into  $k$  numbers of substrings  $w_1, \dots, w_k$  so that there exists a positive integer  $n$  and  $lg(w_i) = nt_i$  for each  $i$ . Also, note that  $G^T$  has  $2 \prod_{i=1}^k m^{t_i} + 1$  number of matrix rules, where  $m$  is the cardinality of an alphabet  $\Sigma$ .

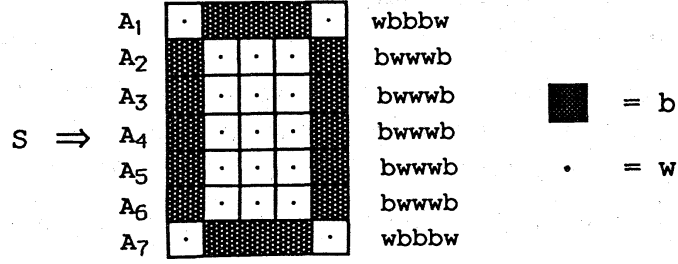


Figure 1: The scheme of encoding the figure '0' into a string

**Theorem 5.2** For any language  $L_t$  over  $\Sigma$ ,  $L_t$  is a  $t$ -even  $k$ -EML if and only if there exists a regular control set  $C$  for the universal  $t$ -even  $k$ -EMG  $G^T$  such that  $L_t = L_C(G^T)$  holds.

*Proof.* By Lemma 5.1, we may assume that any  $t$ -even  $k$ -EML is generated by a  $t$ -even  $k$ -EMG  $G_t$  of the form described in Lemma 5.1. To prove this theorem, we have only to note that  $G^T$  is a primitive  $t$ -even  $k$ -EMG of  $G_t$  and a homomorphism  $h$  and the corresponding NFA to  $G_t$  can be defined in the same way as in the proof of Theorem 3.4.  $\square$

Since the universal  $t$ -even  $k$ -EMG  $G^T$  is unambiguous, by Lemma 4.1, for any  $t$ -even  $k$ -EML  $L_t$  over  $\Sigma$ , a control set  $C$  for  $G^T$  such that  $L_t = L_C(G^T)$  is regular and unique. Moreover,  $C = \{\alpha \mid S \xrightarrow[G^T]{\alpha} w, w \in L_t\}$ . Therefore, since  $G^T$  is fixed for  $\Sigma$ , a learner for  $t$ -even  $k$ -EMLs has only to identify  $C$ . To construct the learner, we have only to prepare a front-end processing algorithm and add it to a learner for regular sets. The algorithm performs three auxiliary tasks described in the section 4. Note that the time complexity of parsing an input string  $w$  in  $G^T$  is bounded by a polynomial of  $n$  and  $m$  [10], where  $n = \lg(w)$  and  $m$  is the cardinality of the alphabet  $\Sigma$ . Therefore, the front-end processing algorithm reduces the problem of identifying an unknown  $t$ -even  $k$ -EML to the problem of identifying an unknown regular control set for a universal  $t$ -even  $k$ -EMG  $G^T$  in polynomial time of sizes of strings, associate words, and DFAs. Hence, we have the following theorem.

**Theorem 5.3** The problem of learning  $t$ -even  $k$ -EMLs is reduced to the problem of learning regular sets. Moreover, if the time complexity of a learner for regular sets is bounded by a polynomial, then that of a learner for  $t$ -even  $k$ -EMLs with using the learner for regular sets is also bounded by a polynomial.

**Example** Equal matrix languages can be considered as digital picture languages introduced in [6]. Figure 1 illustrates how to encode the figure '0' represented on  $7 \times 5$  matrix of rectangular arrays into strings. The figure '0' is encoded into the string  $wb^3w(bw^3b)^5wb^3w$ . Figure 2 shows all figures represented on matrices and their encoded strings. In this case, the set of the figures and finite sequences of them is encoded in  $(1, 1, 1, 1, 1, 1, 1)$ -even 7-EMG and from the results described in this section, we can construct a pattern recognition system which can learn this type of digital pictures.

## 6 Structured Equal Matrix Grammars

A structured  $k$ -EMG is a  $k$ -EMG which displays derivations in corresponding strings. This implies that a learner may use supplemental information on derivations of the  $k$ -EMG.

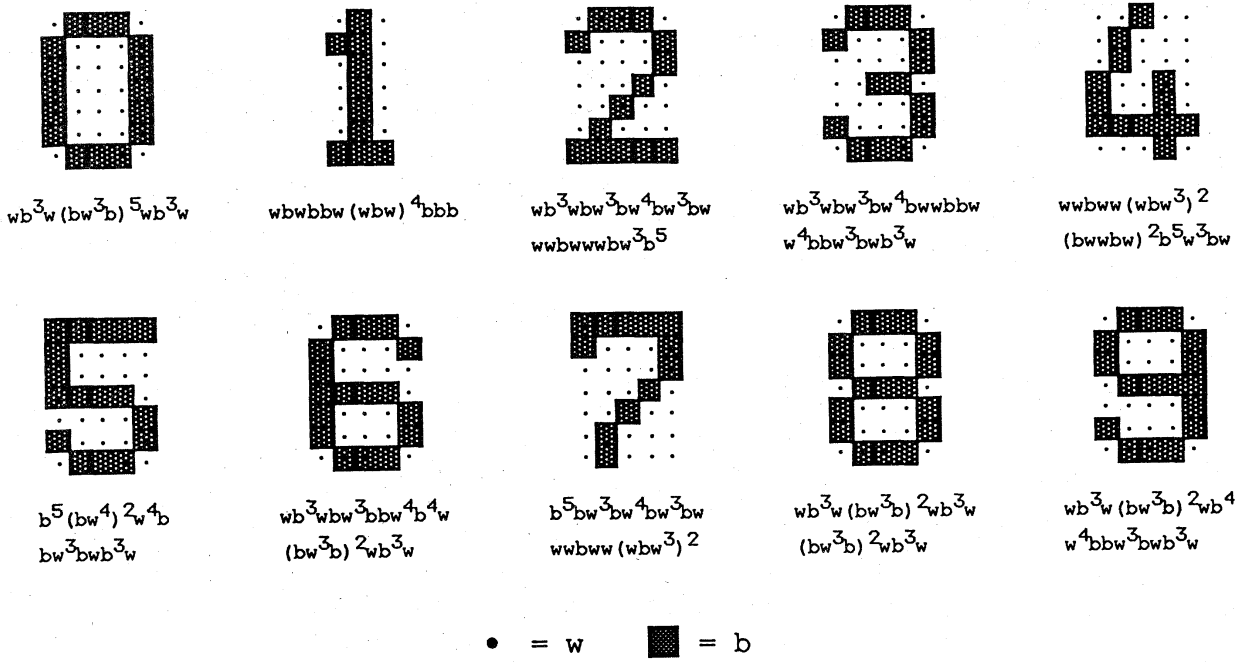


Figure 2: The encoded figures

**Definition** Let  $\#$  be a special symbol not in an alphabet  $\Sigma$ . For a  $k$ -EMG  $G = (N_1, \dots, N_k, \Sigma, \Pi, S)$ , the *structured  $k$ -EMG* of  $G$  is a  $k$ -EMG  $G_s = (N_1, \dots, N_k, \Sigma \cup \{\#\}, \Pi_s, S)$  such that (1)  $\pi_S : [S \rightarrow u_1 A_1 \# \dots u_k A_k \#] \in \Pi_s$  if  $\pi'_S : [S \rightarrow u_1 A_1 \dots u_k A_k] \in \Pi$ , (2)  $\pi_N : [A_1 \rightarrow \# u_1 B_1, \dots, A_k \rightarrow \# u_k B_k] \in \Pi_s$  if  $\pi'_N : [A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k] \in \Pi$ , or (3)  $\pi_T : [A_1 \rightarrow \# u_1, \dots, A_k \rightarrow \# u_k] \in \Pi_s$  if  $\pi'_T : [A_1 \rightarrow u_1, \dots, A_k \rightarrow u_k]$ , where for each  $i$  ( $1 \leq i \leq k$ ),  $A_i, B_i$  are in  $N_i$  and  $u_i$  is in  $\Sigma^*$ .

A *structured  $k$ -EMG* is a structured  $k$ -EMG of some  $k$ -EMG. A *structured  $k$ -EML* is a language generated by a structured  $k$ -EMG. Note that from the definition, for any structured  $k$ -EML  $L_s$ , any element  $w$  in  $L_s$  contains  $kc$  number of  $\#$ s, where  $c$  is a positive integer.

The following lemma can be proved by an induction on the length of strings.

**Lemma 6.1** *Any structured minimal  $k$ -EMG is unambiguous.*

Note that the time complexity of parsing an input string  $w$  in a structured minimal  $k$ -EMG  $G_s^0$  is bounded by a polynomial of  $n$ ,  $p$ , and  $q$  [10], where  $n = \lg(w)$ ,  $p$  is the number of matrix rules of  $G_s^0$ , and  $q$  is the maximum size of the matrix rules of  $G_s^0$ .

Let  $L_s$  be a structured  $k$ -EML. Since any structured minimal  $k$ -EMG is unambiguous, by Lemma 4.1, for any primitive  $k$ -EMG  $G_s^0$  of  $L_s$ , a control set  $C = \{\alpha \mid S \xrightarrow[G_s^0]{\alpha} w, w \in L_s\}$  for  $G_s^0$  is regular and unique. From this, a learner identifies a structured  $k$ -EML  $L_s$  by identifying a primitive  $k$ -EMG  $G_s^0$  of  $L_s$  and a unique regular control set  $C$  for  $G_s^0$ . We next show that  $G_s^0$  can be identified efficiently from the given examples.

Let  $G_s$  be a structured  $k$ -EMG. A string  $w$  is said to *exercise* a matrix rule  $\pi$  of  $G_s$  if and only if there exists a derivation  $S \xrightarrow[G_s]{\alpha} x \xrightarrow[G_s]{\pi} y \xrightarrow[G_s]{\beta} w$ . A matrix rule  $\pi$  of  $G_s$  is said to be *live* if and only if there exists a string  $w$  in  $L(G_s)$  which exercises  $\pi$ .



**Algorithm CRA**

Dividing  $w$  into  $k$  number of substrings  $w_1, \dots, w_k$   
 such that  $w = w_1\# \dots w_k\#$  and each  $w_i$  has the same number of  $\#$ s;  
**For** each  $i$ , divide  $w_i$  into  $u_{i1}, \dots, u_{in}$  such that  $w_i = u_{i1}\# \dots \#u_{in}$  and  $u_{ij} \in \Sigma^*$ ;  
 $\Pi_s := \{ \pi_S : [S \rightarrow u_{11}S_1\# \dots \#u_{k1}S_k\#] \}$   
 $\cup \{ \pi_N : [S_1 \rightarrow \#u_{1j}S_1, \dots, S_1 \rightarrow \#u_{kj}S_k] \mid 2 \leq j \leq n-1 \}$   
 $\cup \{ \pi_T : [S_1 \rightarrow \#u_{1n}, \dots, S_k \rightarrow \#u_{kn}] \};$   
 output  $\Pi_s$  and halt;

Figure 3: An algorithm generating matrix rules

**Definition** A finite subset  $RS(L_s)$  of a structured  $k$ -EML  $L_s$  is said to be a *representative sample* of  $L_s$  if and only if there exists a structured  $k$ -EMG  $G_s$  such that  $L_s = L(G_s)$  and for every live matrix rule  $\pi$  of  $G_s$ , there exists an element in  $RS(L_s)$  which exercises  $\pi$ .

Let  $RS$  be a finite subset of a structured  $k$ -EML  $L_s$ . Given a string  $w$  of  $L_s$ , Algorithm CRA illustrated in Figure 3 outputs a set of matrix rules. Let  $\Pi_s^0$  be the union of the output sets of CRA for  $RS$ . Then from the definition of representative samples we have:

**Lemma 6.2** *If  $RS$  is a representative sample of  $L_s$ , then  $G_s^0 = (\{S_1\}, \dots, \{S_k\}, \Sigma \cup \{\#\}, \Pi_s^0, S)$  is a primitive  $k$ -EMG of  $L_s$ .*

Note that it is obvious from the construction that given a string  $w$  such that  $n = lg(w)$ , the cardinality of the output of Algorithm CRA is bounded by a polynomial of  $n$  and also note that the time complexity of CRA is bounded by a polynomial of  $n$ .

A learner is said to *identify a  $k$ -EMG  $G$  from structural examples* if and only if it identifies the structured  $k$ -EMG of  $G$  from examples. Since any structured minimal  $k$ -EMG is unambiguous, it follows from Proposition 4.2 that for any unknown structured  $k$ -EML  $L_s$ , if a primitive  $k$ -EMG  $G_s^0$  of  $L_s$  is found, then the problem of identifying  $L_s$  is reduced to the problem of identifying a regular set. Therefore, after constructing  $G_s^0$  with Algorithm CRA, the learner has only to identify a regular control set with using a learner for regular sets. Note that the number of matrix rules of  $G_s^0$  is bounded by the number of examples given to Algorithm CRA and the maximum length of any given example, and that the maximum size of the matrix rules of  $G_s^0$  is also bounded by the maximum length of any given example. Hence, we have:

**Theorem 6.3** *Given a learner for regular sets such that its time complexity is bounded by a polynomial, one can construct a learner which identifies any  $k$ -EMG from structural examples in polynomial time.*

**Example** Again, we show an example for digital picture languages. Figure 4 illustrates an encoding of the correspondence between tables and their histograms into equal matrix languages. In this case, the histogram shows its structure with coloring. This correspondence can be encoded in a structured 6-EMG and from the results described in this section, we can construct a pattern recognition system which can learn this type of digital picture processing.

## 7 Learning from Positive Examples

Angluin [2] has introduced *zero-reversible* finite automata and shown that the class of zero-reversible finite automata is learnable from positive examples. A DFA  $M = (K, \Sigma, \delta, q_0, F)$

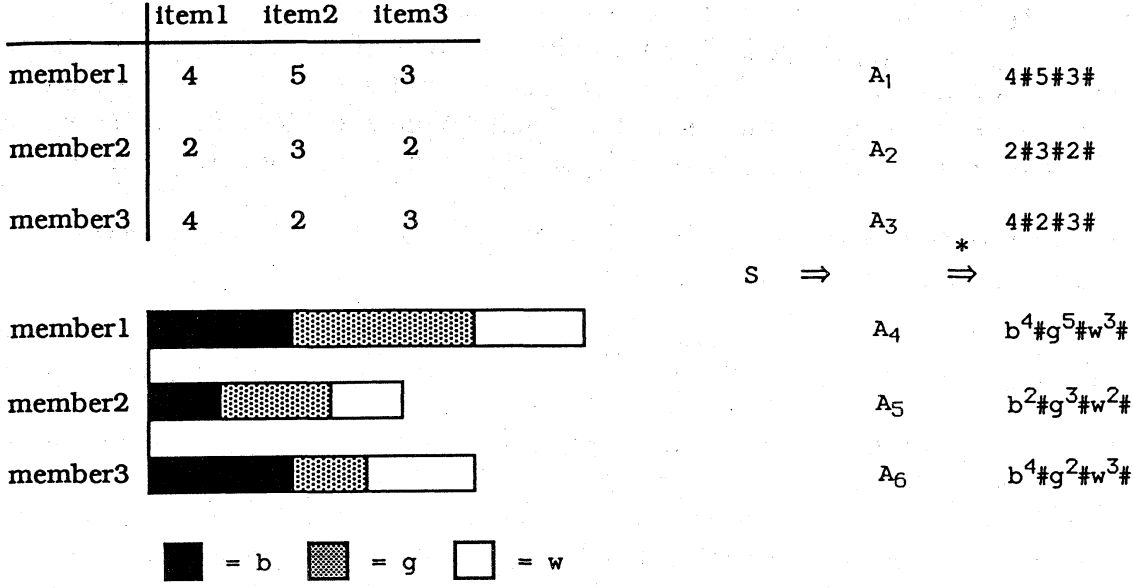


Figure 4: A table and a histogram

is said to be *reset-free* if and only if for no two distinct states  $q_1$  and  $q_2$  do there exist  $a \in \Sigma$  and  $q_3 \in K$  such that  $\delta(q_1, a) = q_3 = \delta(q_2, a)$ . A *zero-reversible finite automata* is a DFA such that it has at most one final state and is reset-free.

A  $k$ -EMG  $G$  is said to be *deterministic* if and only if there are no two distinct matrix rules  $\pi_{S_1} : [S \rightarrow u_1 A_1 \cdots u_k A_k]$  and  $\pi_{S_2} : [S \rightarrow u_1 B_1 \cdots u_k B_k]$  and there are no two distinct matrix rules  $\pi_{N_1} : [A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k]$  and  $\pi_{N_2} : [A_1 \rightarrow u_1 C_1, \dots, A_k \rightarrow u_k C_k]$ . A  $k$ -EMG  $G$  is said to be *reset-free* if and only if there are no two distinct matrix rules  $\pi_1 : [B_1 \rightarrow x_1, \dots, B_k \rightarrow x_k]$  and  $\pi_2 : [C_1 \rightarrow x_1, \dots, C_k \rightarrow x_k]$ .

**Definition** A  $k$ -EMG  $G$  is said to be *in reversible form* if and only if  $G$  is deterministic and reset-free.

**Proposition 7.1** For any  $k$ -EML  $L$ , there exists a  $k$ -EMG  $G$  in the reversible form which generates  $L$ .

*Proof.* Let  $G'$  be a  $k$ -EMG which generates  $L$ . Without loss of generality, we may assume that  $G'$  has no matrix rule of the form  $\pi : [A_1 \rightarrow B_1, \dots, A_k \rightarrow B_k]$ . At first, we introduce new nonterminals  $C_1, \dots, C_k$  and a new matrix rule  $\pi_\lambda : [C_1 \rightarrow \lambda, \dots, C_k \rightarrow \lambda]$  into  $G'$ . Also, we replace each matrix rule of the form  $\pi_T : [A_1 \rightarrow u_1, \dots, A_k \rightarrow u_k]$  by  $\pi_T : [A_1 \rightarrow u_1 C_1, \dots, A_k \rightarrow u_k C_k]$  where each  $u_i \in \Sigma^*$ . Then we construct an NFA corresponding to  $G'$ , convert the NFA into the DFA, and again construct a  $k$ -EMG from the DFA. Let  $G''$  be the modified  $k$ -EMG. It follows from Theorem 3.4 that  $G''$  is deterministic.

Let  $\pi_1 : [B_{1_1} \rightarrow u_1 A_1, \dots, B_{1_k} \rightarrow u_k A_k], \dots, \pi_j : [B_{j_1} \rightarrow u_1 A_1, \dots, B_{j_k} \rightarrow u_k A_k]$  be all  $j$  number of distinct matrix rules of  $G''$  which violate the reset-free condition for  $(A_1, \dots, A_k)$ . Then we introduce  $(j-1)k$  number of new nonterminals  $C_{1_1}, \dots, C_{1_k}, \dots, C_{(j-1)_1}, \dots, C_{(j-1)_k}$  into  $G''$  and replace  $\pi_1, \dots, \pi_j$  by new matrix rules  $\pi'_1 : [B_{1_1} \rightarrow u_1 A_1, \dots, B_{1_k} \rightarrow u_k A_k], \pi'_2 : [B_{2_1} \rightarrow u_1 C_{1_1}, \dots, B_{2_k} \rightarrow u_k C_{1_k}], \dots, \pi'_j : [B_{j_1} \rightarrow u_1 C_{(j-1)_1}, \dots, B_{j_k} \rightarrow u_k C_{(j-1)_k}]$  and

$\pi_1'' : [C_{1_1} \rightarrow A_1, \dots, C_{1_k} \rightarrow A_k], \pi_2'' : [C_{2_1} \rightarrow C_{1_1}, \dots, C_{2_k} \rightarrow C_{2_k}], \dots, \pi_j'' : [C_{(j-1)_1} \rightarrow C_{(j-2)_1}, \dots, C_{(j-1)_k} \rightarrow C_{(j-2)_k}]$ . It is easy to verify that by a finite number of repetitions of this modification the modified  $G''$  is reset-free and generates  $L$ . Let  $G$  be the modified  $k$ -EMG. Also, if  $G''$  is deterministic before the modification, then  $G$  is still deterministic because at each step of repetitions of the modification, we introduce new nonterminals.  $\square$

Since it is obvious from Theorem 3.4 that the corresponding DFA to a  $k$ -EMG in reversible form is a zero-reversible automaton, we have the following theorem:

**Theorem 7.2** *For any  $k$ -EML  $L$ , there exist a minimal  $k$ -EMG  $G^0$  and a zero-reversible automaton  $M$  such that  $L = L_{T(M)}(G^0)$  holds.*

We consider the problem of learning  $k$ -EMLs from positive structural examples. As we have noted in Section 6, for any unknown structured  $k$ -EML  $L_s$ , if a primitive  $k$ -EMG  $G_s^0$  of  $L_s$  is found, then the problem of identifying  $L_s$  is reduced to the problem of identifying a regular set. Also, since a representative sample consists of only positive examples,  $G_s^0$  can be found only from positive examples. These observations imply the following theorem:

**Theorem 7.3** *For any  $k$ -EML, there exist a  $k$ -EMG  $G$  learnable only from positive structural examples.*

Note that since the time complexity of Angluin's learner is bounded by a polynomial, there exists a learner which learns a  $k$ -EMG in reversible form from the given positive structural examples in polynomial time.

## 8 Application to Learning Multitape Automata

$k$ -EMGs are closely related to  $k$ -tape automata. In this section, we apply the learning methods for  $k$ -EMGs described in the above to the problem of learning  $k$ -tape automata.

For a positive integer  $k$ , we denote a  $k$ -tape (nondeterministic) automaton  $M_k$  (over an alphabet  $\Sigma$ ) by a 5-tuple  $(K, \Sigma, \delta, Q, F)$  where  $K$  is a finite set of states,  $\delta$  is a finite subset of  $(k+2)$ -tuples  $K \times \Sigma^{*k} \times K$ ,  $Q \subseteq K$  is the set of initial states, and  $F \subseteq K$  is the set of final states. We extend  $\delta$  in a natural way to  $\delta^*$  in the following way:  $(q, w_1, \dots, w_k, q') \in \delta^0$  if and only if  $q = q'$  and  $w_1 = \dots = w_k = \lambda$ ,  $(q, w_1, \dots, w_k, q') \in \delta^{(n+1)}$  if and only if there exist  $q'', u_1, \dots, u_k, v_1, \dots, v_k$  such that  $w_i = u_i v_i$  for each  $i$ ,  $(q, u_1, \dots, u_k, q'') \in \delta^n$ , and  $(q'', v_1, \dots, v_k, q') \in \delta$ .  $\delta^* = \bigcup_{n \geq 0} \delta^n$ . The set of all  $k$ -tuples accepted by  $M_k$ , denoted  $T(M_k)$ , is the set

$$T(M_k) = \{(w_1, \dots, w_k) \mid \text{there exist } q_0 \in Q \text{ and } q_f \in F \text{ such that } (q_0, w_1, \dots, w_k, q_f) \in \delta^*\}.$$

**Definition** A  $k$ -tape EMG is a  $k$ -EMG  $G_k = (N_1, \dots, N_k, \Sigma \cup \{\$, \Pi_k, S)$  such that  $\$ \notin \Sigma$  and each matrix rule in  $\Pi_k$  is of the form (1)  $\pi_S : [S \rightarrow A_1 \dots A_k]$ , (2)  $\pi_N : [A_1 \rightarrow u_1 B_1, \dots, A_k \rightarrow u_k B_k]$ , or (3)  $\pi_T : [A_1 \rightarrow u_1 \$, \dots, A_k \rightarrow u_k \$]$ , where  $A_i, B_i \in N_i$ ,  $u_i \in \Sigma^*$ .

We show the relation between  $k$ -tape automata and  $k$ -tape EMGs in the following proposition without the formal proof:

**Proposition 8.1** *Let  $T_k$  be a subset of  $\Sigma^{*k}$ . There exists a  $k$ -tape automaton  $M_k$  over  $\Sigma$  such that  $T_k = T(M_k)$  if and only if there exists a  $k$ -tape EMG  $G_k$  such that  $L(G_k) = \{w_1\$ \cdots \$w_k\$ \mid (w_1, \dots, w_k) \in T_k\}$ .*

We note that the time complexity of constructing  $M_k$  from  $G_k$  is bounded by a polynomial of  $p$  and  $q$ , where  $p$  is the number of matrix rules of  $G_k$  and  $q$  is the maximum size of the matrix rules of  $G_k$ .

Now we apply the learning methods for  $k$ -EMGs to the problem of learning  $k$ -tape automata. We define two subclasses of  $k$ -tape automata according to the subclasses of  $k$ -EMLs introduced in the above. A  $k$ -tape automaton  $M_{kt}$  is said to be *t-even* if and only if there exists a  $t$ -even  $k$ -tape EMG  $G_{kt}$  such that  $L(G_{kt}) = \{w_1\$ \cdots \$w_k\$ \mid (w_1, \dots, w_k) \in T(M_{kt})\}$ . A  $k$ -tape automaton  $M_{ks}$  is said to be *structured* if and only if there exists a structured  $k$ -tape EMG  $G_{ks}$  such that  $L(G_{ks}) = \{w_1\$ \cdots \$w_k\$ \mid (w_1, \dots, w_k) \in T(M_{ks})\}$ . Then the problem of learning  $t$ -even  $k$ -tape automata is reduced to the problem of learning regular sets. Given a learner for regular sets whose time complexity is bounded by a polynomial, one can construct learners for  $t$ -even  $k$ -tape automata and structured  $k$ -tape automata which take time polynomial. Moreover, for any  $k$ -tape automaton  $M_k$ , there exists a  $k$ -tape automaton  $M'_k$  such that  $T(M_k) = T(M'_k)$  and  $M'_k$  is learnable from positive structural examples.

## References

- [1] V. Amar and G. Putzolu. Generalizations of regular events. *Information and Control*, 8:56–63, 1965.
- [2] D. Angluin. Inference of reversible languages. *Journal of the ACM*, 29(3):741–765, 1982.
- [3] D. Angluin and C. H. Smith. Inductive inference : Theory and methods. *ACM Computing Surveys*, 15(3):237–269, 1983.
- [4] O. H. Ibarra. Simple matrix languages. *Information and Control*, 17:359–394, 1970.
- [5] Y. Sakakibara. An efficient learning of context-free grammars from positive structural examples. Research Report 93, IIAS-SIS, FUJITSU LIMITED, 1989.
- [6] G. Siromoney, R. Siromoney, and K. Krithivasan. Abstract families of matrices and picture languages. *Computergraphics and Image Processing*, 1:284–307, 1972.
- [7] R. Siromoney. On equal matrix languages. *Information and Control*, 14:135–151, 1969.
- [8] Y. Takada. Grammatical inference for even linear languages based on control sets. *Information Processing Letters*, 28(4):193–199, 1988.
- [9] Y. Takada. Inferring parenthesis linear grammars based on control sets. *Journal of Information Processing*, 12(1):27–33, 1988.
- [10] Y. Takada. Learning equal matrix grammars based on regular control sets. in technical report: FUJITSU IIAS-SIS Workshop on Computational Learning Theory '89, IIAS-SIS, FUJITSU LIMITED, 1990.

- [11] D. Wood. Bounded parallelism and regular languages. In G. Rozenberg and A. Salomaa, editors, *L Systems, Lecture Notes in Computer Science, No.15*, pages 292–301. Springer-Verlag, 1974.